

VAIDS-DSML exercise block 9

Jasmin Duvivié

simple linear regression

- problem

- 1599 samples of wines
- 11 features
- 1 target: quality
- 1 feature as linear regressor: alcohol

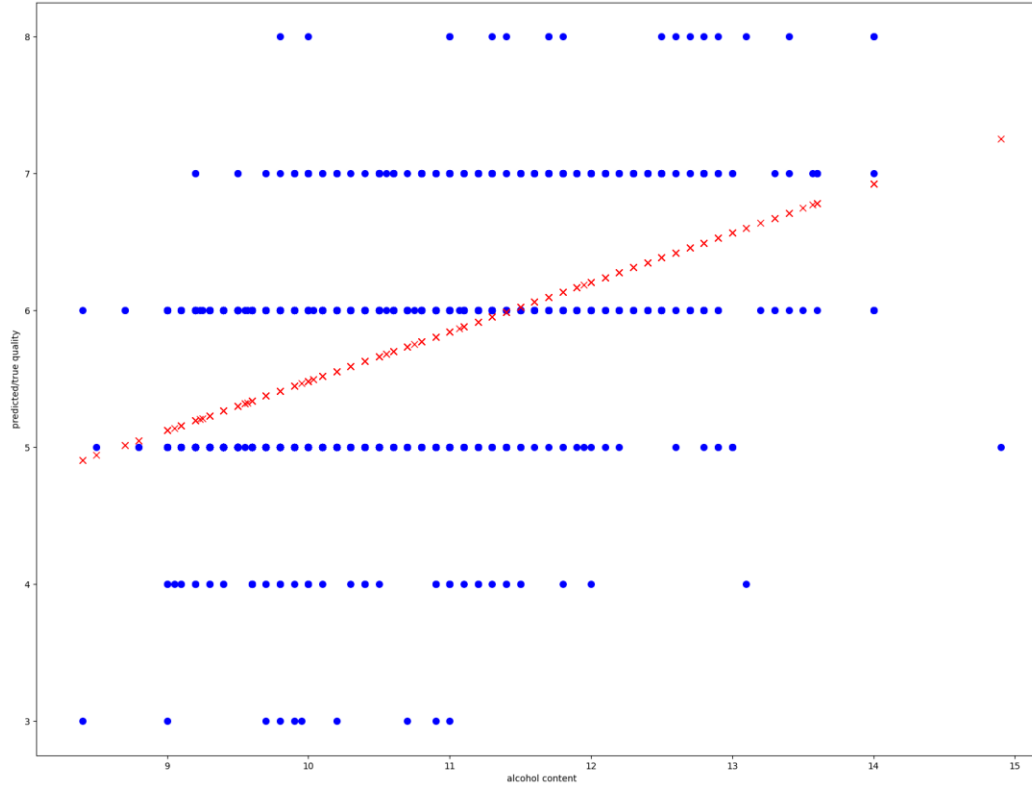
- solution

- use `sklearn.linear_model`
- construct model manually
- plot observed and predicted values for both models
- compare plots and ordinary least squares parameters

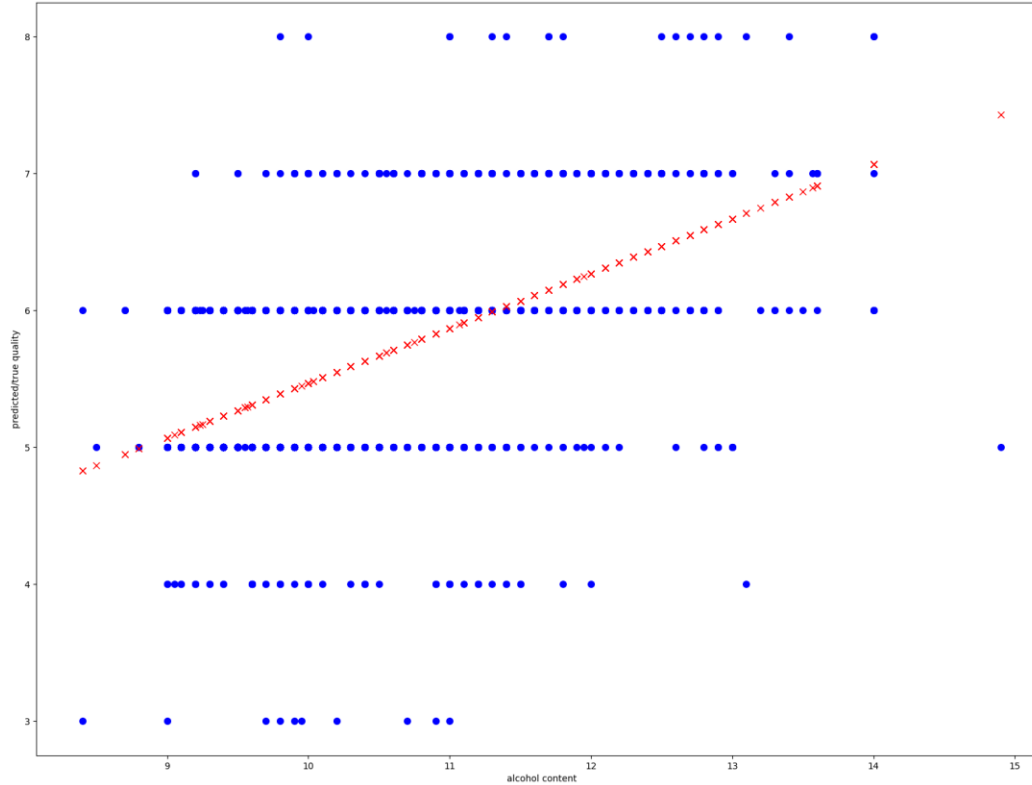
simple linear regression manually

- $\hat{y} = k * x + d$
- calculate \hat{y} for a bunch of different combinations of k and d
- for each combination calculate the sum of squared residuals
- the combination of k and d which results in the smallest sum of squared residuals is used for our model

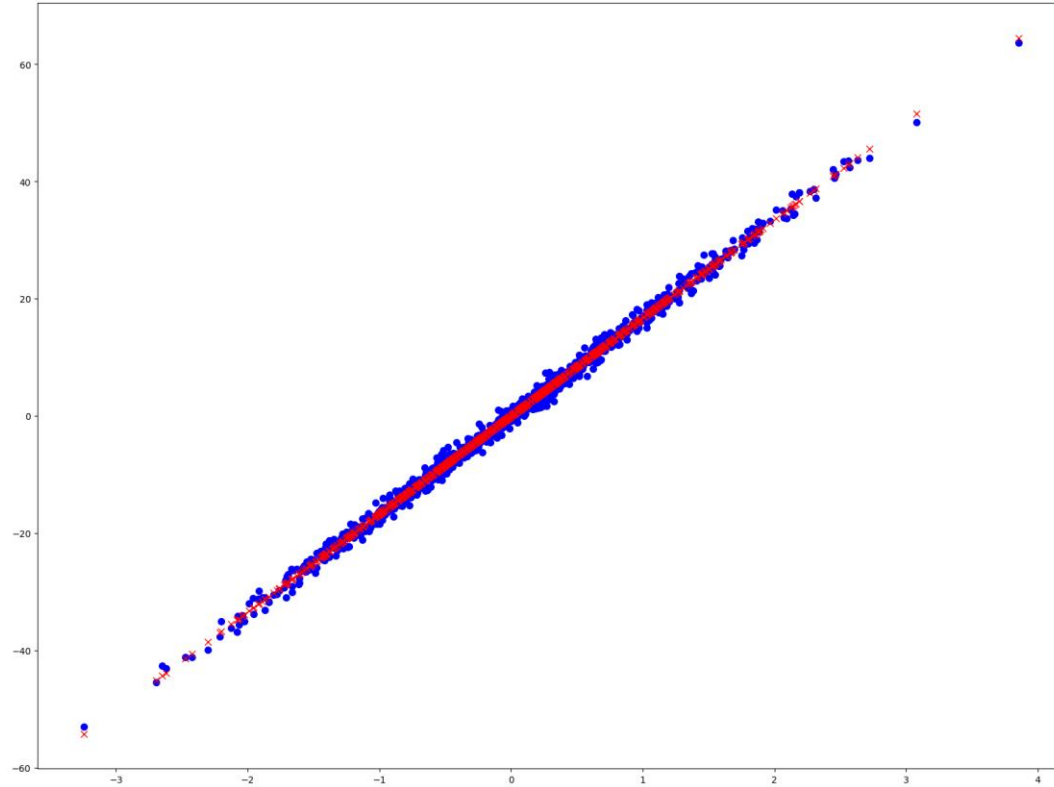
simple linear regression



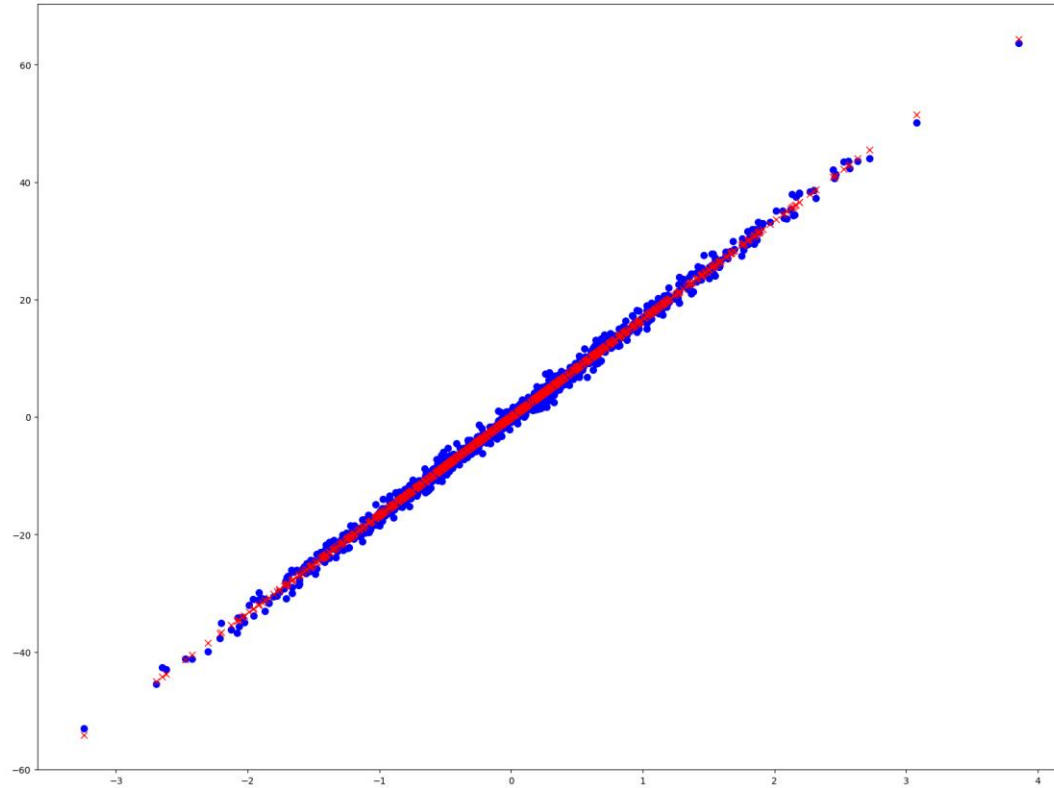
simple linear regression manually



simple linear regression



simple linear regression manually



simple linear regression

```
01 best_intercept = {float64: ()} 0.0100000000000001563
01 best_slope = {float64: ()} 16.7
> coef = polyreg.coef_
01 coef = {float64: ()} 16.73678701
> mse = polyreg.mse_
01 mse = {float64: ()} 0.981807525127997
> mse_man = mse
01 mse_man = {float64: ()} 0.9814997526071848
> coef_ = ndarray: (1,) [16.73678701] ...View as Array
01 coef_ = {float64: ()} 16.73678701
01 mse_man = {float64: ()} 0.9814997526071848
01 intercept_ = {float64: ()} 0.004526205905821257
```

```
01 best_intercept = {float64: ()} 1.47000000000000033
01 best_slope = {float64: ()} 0.4

> # Print the coefficients
> print(best_intercept)
1.47
> print(best_slope)
0.4
> # Print the MSE
> print(mse)
0.503984025671457
> print(mse_man)
0.5057343548050866

> # Print the coefficients as a numpy array
> coef_ = np.array([best_intercept, best_slope])
> print(coef_)
[ 1.47  0.4]
> # Print the intercept and slope
> print(intercept_, slope_)
1.47 0.4
```